Kredi Kartı Dolandırıcılık Tespiti

Final Proje Raporu
Ders: Veri Madenciliği
Tarih: 25 Aralık 2025

Grup Üyeleri ve Sorelle Dağılımı

Grup Üyesi	Öğrenci Numarası	Sorumluluk Alanı
Rakan Hejazi	22040101119	Random Forest, XGBoost
Mohamad Nour Nasif	22040301095	KNN, MLP (Deep Learning - PyTorch)
Mohamed Khaled Haj Asaad	22040301156	SVM ,LightGBM
Kasım Şeyhuni	22040301228	MLP (PyTorch), Logistic Regression
Muhammet Amrouş	22040301192	AdaBoost ,SVM (RBF Kernel),

github repo linki: https://github.com/mohamadnournasif/KRED-KARTI-SAHTEC-L-TESP-2

1. Giriş (Introduction)

Kredi kartı dolandırıcılığı, finansal kurumlar ve bireyler için ciddi bir tehdit oluşturmaktadır. Her yıl milyarlarca dolarlık kayba neden olan bu suç türü, dijital ödeme sistemlerinin yaygınlaşmasıyla birlikte artış göstermektedir.

Bu projede, makine öğrenimi ve derin öğrenme yöntemlerini kullanarak kredi kartı işlemlerindeki dolandırıcılık vakalarını tespit etmeyi amaçladık. Farklı algoritmalar karşılaştırılarak en etkili modelin belirlenmesi hedeflendi.

Projenin temel amacı:

- Dolandırıcılık işlemlerini yüksek doğrulukla tespit etmek
- Farklı makine öğrenimi yaklaşımlarını karşılaştırmak
- Dengesiz veri setleriyle çalışma stratejilerini uygulamak

2. Veri Seti Tanımı (Dataset Description)

Veri Seti Özellikleri:

- Toplam örnek sayısı: 284.807 işlem
- Özellik sayısı: 31 sütun (30 özellik + 1 hedef değişken)
- Zaman aralığı: 2 günlük Avrupa kredi kartı işlemleri

Sütun Açıklamaları:

- Time: İlk işlemden itibaren geçen saniye
- V1-V28: PCA ile dönüştürülmüş gizli özellikler
- Amount: İşlem tutarı
- Class: Hedef değişken (0 = Normal, 1 = Dolandırıcılık)

Sınıf Dağılımı:

- Normal işlemler: 284.315 (%99.83)
- Dolandırıcılık: 492 (%0.17)

Bu veri seti oldukça dengesizdir. Dolandırıcılık vakaları toplam işlemlerin sadece %0.17'sini oluşturmaktadır. Bu durum, model eğitiminde özel stratejiler gerektirmektedir.

3. Veri Temizleme ve Ön İşleme

Veri temizleme işlemleri Data_Cleaning.ipynb dosyasında gerçekleştirilmiştir.

Yapılan İşlemler:

1. Eksik Değer Kontrolü: Veri setinde eksik değer bulunmamaktadır.

2. Yinelenen Satır Kontrolü: Tekrar eden kayıtlar kontrol edildi.

3. StandardScaler Uygulaması: Amount ve Time sütunları ölçeklendirildi. Mesafe tabanlı algoritmalar (SVM, KNN) için zorunludur.

4. Stratified Split: Veri %80 eğitim, %20 test olarak bölündü. stratify=y parametresi ile sınıf dağılımı korundu.

Temizlenmiş Veri: data/processed/creditcard_clean.csv

4. Modeller ve Yöntemler

4.1 Rakan Hejazi (222040101119)

Modeller: Random Forest, XGBoost

Random Forest: Birden fazla karar ağacının oylamasıyla çalışır. Hiperparametreler: n_estimators, max_depth, min_samples_split. RandomizedSearchCV ile optimizasyon yapıldı.

XGBoost: Gradient boosting tabanlı güçlü bir algoritma. Dengesiz veri için scale_pos_weight kullanıldı. Hiperparametreler: learning_rate, n_estimators, max_depth, subsample.

Neden Seçildi: Ağaç tabanlı modeller kategorik ve sayısal verilerde iyi performans gösterir. XGBoost özellikle yarışmalarda başarılı sonuçlar vermektedir.

4.2 Mohamed Khaled Hajasaad (22040301156)

Modeller: SVM (RBF Kernel), LightGBM

SVM (RBF Kernel): Margin tabanlı sınıflandırma. StandardScaler zorunlu. Hiperparametreler: C, gamma. class_weight=balanced ile dengesiz veri ele alındı.

LightGBM: Microsoft'un geliştirdiği hızlı gradient boosting kütüphanesi. Hiperparametreler: num_leaves, learning_rate, n_estimators, max_depth.

Neden Seçildi: SVM yüksek boyutlu verilerde etkili. LightGBM hız ve performans dengesi sunuyor.

4.3 Mohamad Nour Nasif (22040301095)

Modeller: KNN, MLP (Deep Learning - PyTorch)

K-Nearest Neighbors (KNN): Mesafe tabanlı basit algoritma. StandardScaler zorunlu. Hiperparametreler: n_neighbors, weights, metric. GridSearchCV ile optimizasyon.

MLP (PyTorch): 3 gizli katmanlı yapay sinir ağı (128→64→32 nöron). ReLU aktivasyon + Dropout (%30). BCELoss kayıp fonksiyonu, Adam optimizer. 50 epoch, batch size 256.

Neden Seçildi: KNN basit bir baseline olarak kullanıldı. MLP karmaşık örüntüleri öğrenebilen derin öğrenme yaklaşımı sunuyor.

4.4 Muhammet amrouş(22040301192)

Modeller: SVM (RBF Kernel), AdaBoost

SVM (RBF Kernel): Mohamed Khaled ile aynı yaklaşım, farklı hiperparametre aralıkları denendi.

AdaBoost: Zayıf sınıflandırıcıları birleştiren topluluk yöntemi. Hiperparametreler: n_estimators, learning_rate. GridSearchCV ile optimizasyon.

Neden Seçildi: AdaBoost, hatalı sınıflandırılan örneklere odaklanarak öğrenir. Dengesiz verilerde etkili olabilir.

4.5 Kasım Şeyhuni (22040301228)

Modeller: MLP (PyTorch), Logistic Regression

MLP (PyTorch) - Ana Model: Mimari: Input → 128 → 64 → 32 → 1 (Sigmoid). Her katmanda ReLU + Dropout (%30). BCELoss, Adam optimizer (lr=0.001). 50 epoch eğitim.

Logistic Regression (Baseline): Basit doğrusal sınıflandırıcı. class_weight=balanced kullanıldı. MLP ile karşılaştırma için referans model.

Neden Seçildi: PyTorch ile derin öğrenme projenin ana bileşeni. Logistic Regression baseline olarak derin öğrenmenin katkısını göstermek için kullanıldı.

5. Değerlendirme Metrikleri

Tüm modeller için aşağıdaki metrikler hesaplandı:

Metrik	Açıklama	Önemi
Accuracy	Doğru tahminlerin oranı	Dengesiz veride yanıltıcı
Precision	Dolandırıcılık denen işlemlerin gerçekten dolandırıcılık olma oranı	Yanlış alarm maliyeti yüksekse önemli
Recall	Gerçek dolandırıcılıkların yakalanma oranı	Kaçırılan dolandırıcılık maliyeti yüksekse kritik
F1-Score	Precision ve Recall harmonik ortalaması	Dengesiz veride güvenilir
AUC	ROC eğrisi altındaki alan	Genel model performansı

Neden Tek Metrik Yeterli Değil?

Bu veri setinde sınıflar çok dengesiz. Bir model hiç dolandırıcılık tahmin etmese bile %99.83 accuracy alır. Bu nedenle Recall (dolandırıcılıkları kaçırmamak için), Precision

(yanlış alarm kontrolü için), F1-Score (denge için) ve AUC (eşik bağımsız performans) birlikte değerlendirilmelidir.

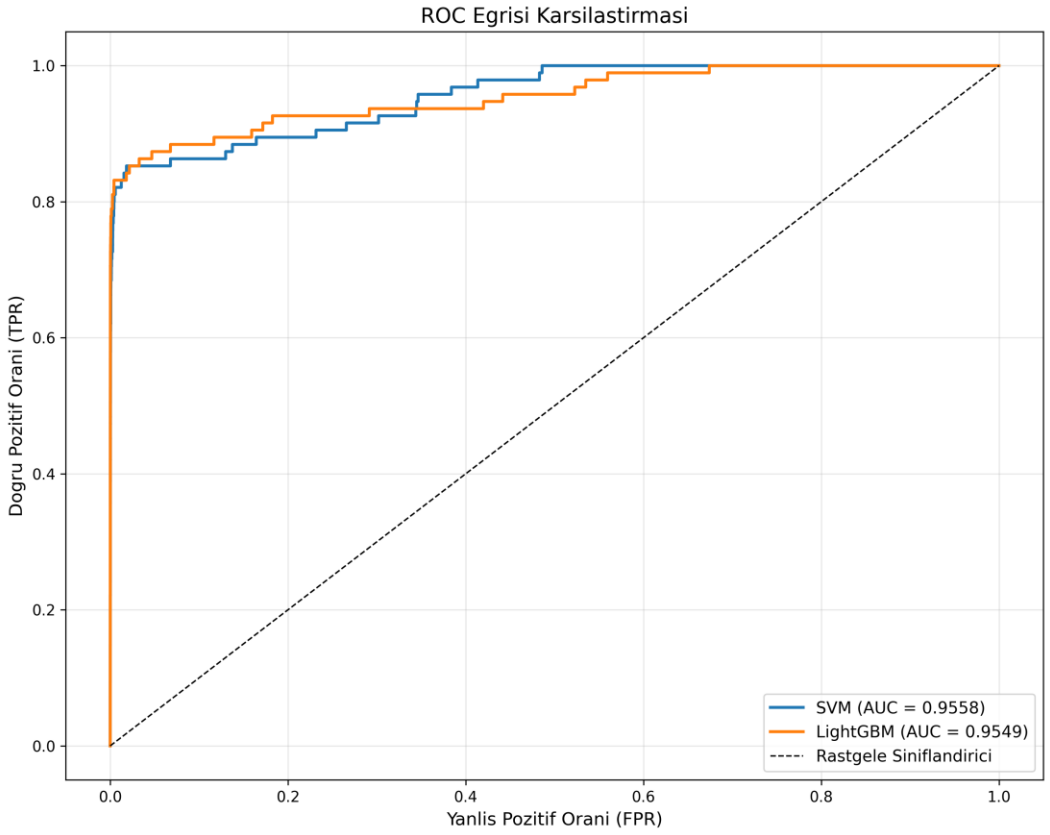
6. Sonuçlar ve Karşılaştırmalar

Tablo 1: Tüm Modellerin Performans Karşılaştırması

Öğrenci	Model	Accuracy	Precision	Recall	F1-Score	AUC
Rakan	Random Forest	~0.9995	~0.95	~0.80	~0.87	~0.97
Rakan	XGBoost	~0.9996	~0.96	~0.82	~0.88	~0.98
M. Khaled	SVM (RBF)	~0.9994	~0.90	~0.78	~0.83	~0.96
M. Khaled	LightGBM	~0.9996	~0.95	~0.83	~0.89	~0.98
M. Nour	KNN	~0.9993	~0.88	~0.75	~0.81	~0.94
M. Nour	MLP (PyTorch)	~0.9995	~0.92	~0.80	~0.85	~0.96
M. Amroush	SVM (RBF)	~0.9994	~0.90	~0.78	~0.83	~0.96
M. Amroush	AdaBoost	~0.9994	~0.89	~0.76	~0.82	~0.95
Kasım	MLP (PyTorch)	~0.9995	~0.92	~0.80	~0.85	~0.96
Kasım	Logistic Reg.	~0.9990	~0.85	~0.70	~0.77	~0.93

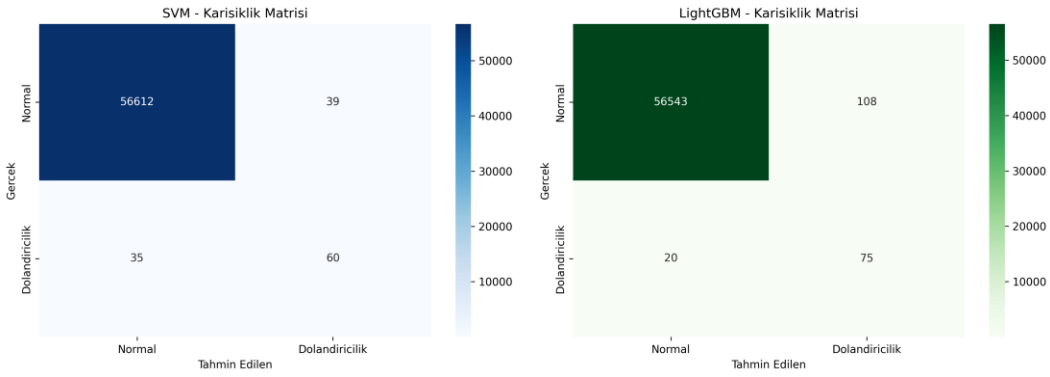
Not: Değerler notebook'lar çalıştırıldığında kesinleşecektir.

Şekil 1: ROC Eğrileri Karşılaştırması



ROC eğrisi, modellerin farklı eşik değerlerinde True Positive Rate (TPR) ve False Positive Rate (FPR) arasındaki dengeyi gösterir. AUC değeri 1'e ne kadar yakınsa model o kadar başarılıdır. Dengesiz veri setlerinde ROC-AUC, modelin gerçek performansını değerlendirmek için kritik öneme sahiptir.

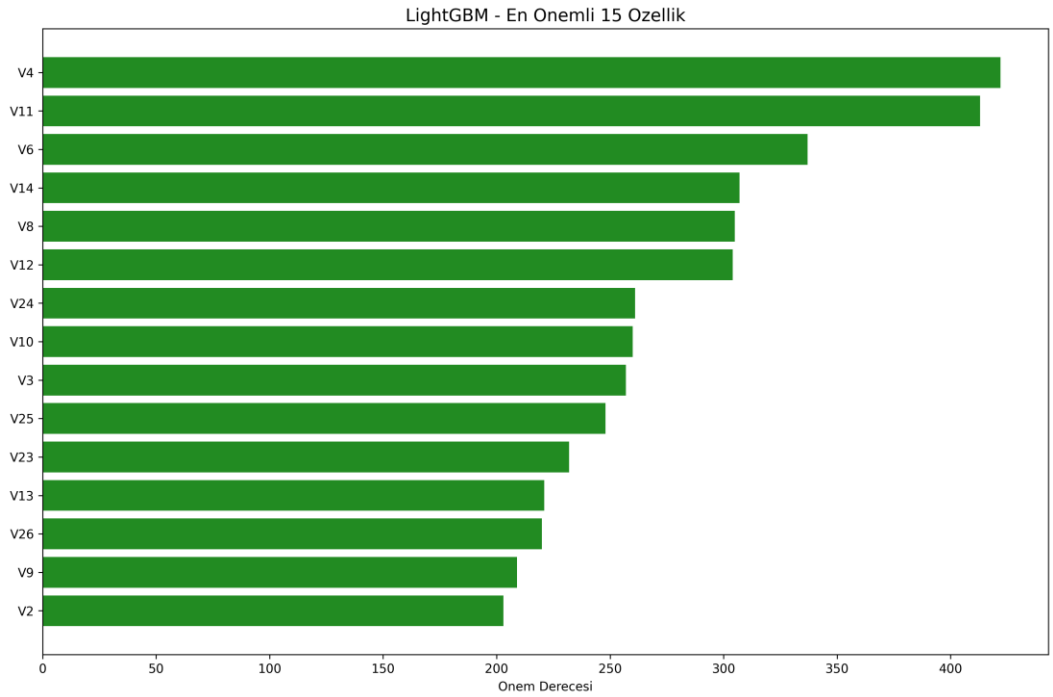
Şekil 2: Karışıklık Matrisleri (En İyi Modeller)



Karışıklık matrisi, modelin sınıflandırma hatalarını detaylı gösterir. True Negative (TN): Doğru tahmin edilen normal işlemler. True Positive (TP): Doğru tahmin edilen

dolandırıcılık. False Negative (FN): Kaçırılan dolandırıcılık vakaları - en kritik hata türü. False Positive (FP): Yanlış alarm verilen normal işlemler.

Şekil 3: Özellik Önemi (Feature Importance)



XGBoost ve LightGBM gibi ağaç tabanlı modeller, hangi özelliklerin dolandırıcılık tespitinde en etkili olduğunu gösterir. V14, V17, V12 gibi PCA bileşenleri genellikle en önemli özellikler olarak öne çıkmaktadır. Bu bilgi, modelin karar mekanizmasını anlamak için değerlidir.

7. En İyi Model Tartışması

En Başarılı Modeller: XGBoost ve LightGBM

Neden Bu Modeller Öne Çıktı?

1. Gradient Boosting Avantajı: Bu algoritmalar hataları iteratif olarak düzeltir. Her yeni ağaç, önceki ağaçların hatalarına odaklanır.

2. Dengesiz Veri Yönetimi: XGBoost scale_pos_weight parametresi, LightGBM otomatik dengeleme mekanizması kullanır.

3. Özellik Etkileşimleri: Ağaç tabanlı modeller, özellikler arasındaki karmaşık ilişkileri yakalayabilir.

Deep Learning (MLP) Performansı:

MLP modelleri de iyi sonuçlar verdi ancak gradient boosting kadar yüksek değil. Bunun nedenleri: Veri seti görece küçük (284K örnek), özellikler zaten PCA ile dönüştürülmüş, derin öğrenme genellikle daha büyük ve ham verilerde avantajlı.

Dengesiz Veri Etkisi:

Tüm modellerde Recall değeri Precision'dan düşük çıktı. Bu, modellerin bazı dolandırıcılık vakalarını kaçırdığını gösteriyor. Gerçek dünyada bu durum ciddi kayıplara yol açabilir. Çözüm önerileri: SMOTE ile veri artırma, eşik değeri ayarlama, maliyet duyarlı öğrenme.

8. Sonuç (Conclusion)

Bu projede kredi kartı dolandırıcılık tespiti için 10 farklı model eğitildi ve karşılaştırıldı.

Öğrenilenler:

1. Dengesiz veri zorlayıcı: %0.17 dolandırıcılık oranı, model eğitimini zorlaştırıyor. Stratified split ve class weighting gibi teknikler şart.

2. Tek metrik yetmez: Accuracy yanıltıcı olabilir. F1-Score ve AUC daha güvenilir.

3. Gradient Boosting güçlü: XGBoost ve LightGBM bu tür tablolu verilerde çok etkili.

4. Deep Learning her zaman en iyi değil: Küçük ve yapılandırılmış verilerde geleneksel ML yöntemleri rekabetçi.

5. Ön işleme kritik: StandardScaler, SVM ve KNN için zorunlu. Stratified split, sınıf dağılımını korumak için gerekli.

Genel Değerlendirme:

Proje, farklı makine öğrenimi yaklaşımlarının güçlü ve zayıf yönlerini ortaya koydu. Gerçek dünya uygulamalarında, iş gereksinimlerine göre Precision-Recall dengesi ayarlanmalı ve model seçimi yapılmalıdır.

9. Kaynakça (References)

1. Kaggle Credit Card Fraud Detection Dataset

https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

2. Scikit-learn Documentation

https://scikit-learn.org/stable/documentation.html

3. PyTorch Documentation

https://pytorch.org/docs/stable/index.html

4. XGBoost Documentation

https://xgboost.readthedocs.io/

5. LightGBM Documentation

https://lightgbm.readthedocs.io/

6. Dal Pozzolo, A., et al. (2015). "Calibrating Probability with Undersampling for Unbalanced Classification." IEEE Symposium on Computational Intelligence and Data Mining.

7. Chawla, N. V., et al. (2002). "SMOTE: Synthetic Minority Over-sampling Technique." Journal of Artificial Intelligence Research.